第 39 卷第 9 期
录用定稿
计算机应用研究
Application Research of Computers
ChinaXiv合作期刊
Vol. 39 No. 9
Accepted Paper

# SDN 中基于 MS-KNN 算法的 LFA 攻击检测方法 *

孙文悦，王昌达†
(江苏大学 计算机科学与通信工程学院，江苏 镇江 212013)

**摘　要：**针对一种新型的 DDoS 攻击—链路泛洪攻击(link-flooding attack，LFA)难以检测的问题，提出了 SDN 中基于 MS-KNN(Mean Shift- K-NearestNeighbor)方法的 LFA 检测方法。首先通过搭建 SDN 实验平台，模拟 LFA 并构建 LFA 数据集；然后利用改进的加权欧氏距离均值漂移(Mean Shift，MS)算法对 LFA 数据集进行分类；最后利用 K 近邻(K-nearestneighbor，KNN)算法判断分类结果中是否具有 LFA 数据。实验结果表明，相较于 KNN 算法，利用 MS-KNN 不仅得到了更高的准确率，同时也得到了更低的假阳性率。

**关键词：**链路泛洪攻击；SDN；均值漂移算法；K 近邻算法；MS-KNN
**中图分类号：**TP393.08　　doi: 10.19734/j.issn.1001-3695.2022.01.0058

## LFA attack detection method based on MS-KNN algorithm in SDN

Sun Wenyue, Wang Changda†
(*School of Computer & Communication*, *Jiangsu University*, *Zhenjiang Jiangsu* 212013, *China*)

**Abstract:** Abstarct: To address the problem that a new type of DDoS attack, link-flooding attack (LFA) , is difficult to detect, an LFA detection method based on MS-KNN (Mean Shift-K-NearestNeighbor) method in SDN is proposed. Firstly, this paper simulated LFA and constructed LFA dataset by building an SDN experiment platform; secondly, an improved weighted Euclidean distance mean shift (MS) algorithm was used to classify the LFA dataset; finally, the K-nearestneighbor (KNN) algorithm was used to determine whether LFA data were included in the classification results. The experimental results show that the use of MS-KNN not only obtains a higher accuracy rate but also a lower false positive rate compared with the KNN algorithm.

**Key words:** LFA; SDN; MS; KNN; MS-KNN

## 0　引言

分布式拒绝服务攻击(Distributed Denial of Service, DDoS)是网络安全面临的最严重威胁之一。根据绿盟科技携手中国电信发布的《2021 DDoS 攻击态势报告》[1]显示，DDoS 攻击方式复杂多变，令人防不胜防，2021 年 DDoS 混合攻击大幅增长，较 2020 年增长了 80.8%。近年来，一种新的 DDoS 攻击方式——链路泛洪攻击(link-flooding attack, LFA)被引入[2,3]，这种攻击可以有效地切断目标单位(如大学校园、军事基地、一组能源分配站)的互联网连接。

与传统的 DDoS 攻击不同，LFA 攻击者不是直接向目标服务器发送大量的攻击流量，而是通过一群傀儡机(或僵尸网络)攻击一个特定的链路或区域，其目的是阻断该区域内目标主机与外界网络的连接。为达到此目的，傀儡机会向目标区域的诱饵服务器或机器人发送合法、低速率的流量。当流量通过连接这些服务器的关键链路时，目标链接会因链路拥塞而被阻断。这种类型攻击最显著的特点是，它使用合法和低速率的流量来实现重大的性能影响，从而使其特别难以检测、防御[4]。

LFA 有两种攻击类型，Coremelt 攻击[2]和 Crossfire 攻击[3]。其中，Coremelt 攻击由 Studer 等人[2]首先提出，它定义了目标链路，攻击者使用一组相互发送数据的傀儡机来淹没目标链路。发动一个 Coremelt 攻击有 3 个步骤：1)选择网络中的核心链路作为目标链路；2)确定哪些傀儡机可以生成穿越目标链接的流量；3)在步骤 1 中确定的目标链路上发送流量，使目标链路过载。Crossfire 攻击是 Coremelt 攻击的升级版本，它针对于更加复杂的网络，即同时攻击复杂网络中的多个目标链路。这种攻击使用傀儡机作为源，使用一些公共服务器作为目的地，并通过源和目的地之间的通信淹没目标链接。为了确保攻击的持久性，对手可以动态地改变目标链接的集合，这是 Coremelt 攻击所不具备的。发动一个 Crossfire 攻击有 4 个步骤：1)构建链接地图；2)计算流密度并选择目标链接进行攻击设置；3)协调傀儡机分配攻击流并淹没目标链接；4)滚动攻击。

LFA 攻击流量容易隐藏在正常的网络流量中，为了在大量数据中发现 LFA 攻击，需要一个有效的方法。

软件定义网络(Software-Defined Networking, SDN)是一种新兴的网络范式，以粒度、灵活性和弹性为特点，为防御网络攻击提供了新的思路。SDN 通常有三个基本特征[5]:

a)控制平面和数据平面的明确分离，在控制平面作出转发决策。

b)将网络逻辑从硬件实现抽象到软件实现。

c)使用控制器或网络操作系统[6]，实现设备的转发决策。

SDN 提供了一种主流的网络管理架构，该架构摆脱了硬件限制，将网络中的控制平面和数据平面解耦。控制平面可以通过统一的接口协议(如 OpenFlow[7]，P4[8])对网络设备进行管理，并规划转发规则来定义网络策略；数据平面根据定义的规则进行处理、转发数据包等工作。因此，与传统网络结构相比，SDN 最大的区别在于它具有通过操作流表来灵活定义网络设备的转发能力，其转发决策是基于流而不是基于目的地的，并由数据包包头中的字段值定义匹配标准和一组操作。在 SDN 中，流是发送方和接收方设备之间的数据包序

---

**收稿日期：**2022-01-19；**修回日期：**2022-04-01　**基金项目：**国家自然科学基金项目(62072217，61672269)
**作者简介：**孙文悦(1997-)，男，江苏盐城人，硕士研究生，主要研究方向为网络安全；王昌达(1971-)，男(通信作者)，江苏镇江人，教授，博导，博士，主要研究方向为网络安全、信息安全、计算机网络等(changda@ujs.edu.cn).

列。这种基于流的抽象统一了各种类型网络设备的行为，如交换机、路由器、防火墙和中间盒[9]。

SDN 有几个优点[10]，如全网视图、逻辑集中控制、基于软件的流量分析和转发规则的动态更新，这些都可以用于更有效的攻击检测。因此，SDN 架构为检测 LFA 提供了良好的平台。

文献[11]提出了 RL-Shield 来缓解 LFA。RL-Shield 利用 Dirichlet 分布和贝叶斯统计量监测源 IP 行为，并使用 hopby-hop 技术连接相关的节点对，通过频繁地改变路径来实现检测过程。然而，该方法受网络拓扑的影响，在真实网络中可移植性差。文献[12]提出了一种基于混合 SDN 的新机制 BALANCE。BALANCE 通过使用基于服务的混合 SDN，通过在网络中放置节点，使得控制器能够统计网络中所有链路的数据，从而实现拥塞检测。然而，在大型网络中，统计所有链路的流量使得计算开销非常庞大。文献[13]提出了一种新型 LFA 防御系统 LFADefender。该系统基于 SDN 的目标链路选择算法，通过 SDN 控制器查看全局视图，动态地跟踪网络中的流动路径，利用探测器发送大量实时探测分组来检测链路拥塞。然而，该系统反映速度不快，攻击者可能在防御机制生效之前更快地改变目标链接。文献[14]提出了基于损伤概率的 LFA 检测。度量的计算考虑网络中所有节点对之间的最短路径，在庞大的网络中需要一个巨大的计算周期。文献[15]提出了 LinkScope。LinkScope 使用逐跳和端到端网络测量方法检测 LFA 攻击，但是它需要部署许多额外的探测点，探测点必须跨网络部署，这在巨大的网络中资源开销大。探测进度取决于前一天的网络流量，这会在网络中引入额外的流量。文献[16]提出了 Woodpecker，其使用增量 SDN 部署来缓解 LFA。拥塞检测模块应安装在所有 SDN 节点上，但是所有节点可能不具备可编程特性和安装模块的内存。同样，文献[17]提出了软件定义蜜网。充分利用了 SDN 全局视图的优势来推断连接蜜网的潜在蜜节点，增加了攻击者的攻击成本。然而，该方案没有考虑到属性的重要性，而这些属性可以准确定位现实世界基础设施中的瓶颈，并且缺乏额外的图形指标来智能选择蜂蜜节点。

虽然 SDN 架构在网络安全方面的创新具有一些明显的好处，被认为适用于当今网络的高带宽和动态环境[18~20]，但是由于 LFA 的特性，仅仅使用 SDN 架构并不足以检测 LFA。

KNN(K Nearest Neighbor)[21]，即 K-近邻算法，是一种惰性学习法。其基本思想为：先计算待分类样本与已知类别的训练样本之间的距离或相似度，找到距离或相似度与待分类样本数据最近的 k 个邻居，再根据这些邻居所属的类别来判断待分类样本数据的类别。所以对于网络数据的分类，KNN 具有较高的准确度与精确度。然而，KNN 算法针对大数据的分类问题，存在如下缺点：1)对每一个待分类的文本都要计算它到全体已知样本的距离，才能求得它的 k 个最近邻点，而面对大量的网络数据，计算全体样本间的距离会产生巨大的开销；2)在决定测试样本的类别时，该算法只计算最近邻的样本，而当面临大规模网络数据时，数据之间特征的不明显会使分类结果产生偏差。

聚类算法是一种无监督学习算法，其主要功能是降维。LFA 攻击流量容易隐藏在正常的网络流量中，为了在大量数据中有效发现 LFA 攻击的存在，需要聚合相似数据，减少需要分析的数据量。

Mean Shift (MS)算法[22]是一种基于无监督学习的聚类算法，不需要预先提供聚类中心的数量。网络数据可以分为一个或多个集群，并通过分析该集群的特征来识别 LFA 的存在。但 MS 算法没有考虑数据的各种属性对分类的贡献程度，导致聚类结果不满意。因此，如何针对某类数据设置权值对聚类效果有显著影响。

本文为了解决 KNN 算法针对大规模网络数据的缺点，以及 MS 算法分类效果不明显的问题，提出了 MS-KNN 算法。实验结果表明，MS-KNN 不仅克服了不同类型数据分类效果不明显的问题，同时也大大减小了计算时间开销。本文的主要贡献如下：a)提出了一种基于加权欧氏距离的 MS 聚类算法，并将其用于 LFA 网络流量的分类，解决了 MS 算法对于 LFA 流量分类不明显的问题；b)提出了基于 MS-KNN 算法的 LFA 检测方法，将改进的 MS 算法和 KNN 算法相结合，以改进的 MS 算法的输出作为 KNN 算法的输入，通过网格搜索和交叉验证寻找最优参数得到最优检测结果；c)在真实的 SDN 环境下进行实验，相关数据证实了 MS-KNN 的有效性，且相较于传统的 KNN 算法，MS-KNN 算法具有更高的召回率(True Positive Rate，TPR)、精确率(Positive predictive value，PPV)、准确率(Accuracy，ACC)和更低的假阳性率(False Positive Rate，FPR)。

## 1　MS-KNN 算法

### 1.1　MS 算法原理

MS 算法[22]利用概率密度的梯度来寻找局部最优。通过定义核函数，使得随着样本与被偏移点的距离不同，其偏移量对均值偏移向量的贡献不同。最典型的核函数为高斯核与 Epanechnikov 核[23]。通过定义权重系数，使得不同样本点的重要性不一样，由此扩展了 MS 的应用范围。

给定个 $n$ 数据点，分布在 $d$ 维欧式空间 $R^d$，有多变量核密度估计的核 $K(x)$，对称正定带宽矩阵 $\boldsymbol{H}$，在点 $x$ 得到的核密度估计为[24]：

$$f(x)=\frac{1}{nh^d}\sum_{i=1}^{n}K\left(\frac{x-x_i}{h}\right) \tag{1}$$

其中，核函数 $K(x)$ [24]满足

$$K(x)=C_{k,d}\,k\left(\|x\|^2\right) \tag{2}$$

其中，$C_{k,d}$ 是一个标准化常数，它保证 $K(x)$ 积分到 1。核密度估计在式(1)中的梯度为

$$\nabla f_{h,k}(x)=\frac{2C_{k,d}}{nh^{d+2}}\left[\sum_{i=1}^{n}g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)\right]\left[\frac{\sum_{i=1}^{n}x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n}g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}-x\right] \tag{3}$$

均值漂移向量由式(3)得出

$$m_h(x)=\frac{\sum_{i=1}^{n}x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n}g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}-x \tag{4}$$

均值漂移矢量总是指向密度最大增长的方向。均值移动过程是由后续过程迭代形成的：

a)计算均值漂移向量 $m_h(x)$。

b)转换新的中心点 $y_{i+1}=m_h(x)+x$。

MS 算法使用的欧氏距离将数据属性之间的差别等同看待，这一点不能满足实际要求。因此，根据数据属性的重要性，赋予不同的权重，使欧式距离优化为加权欧式距离，以提高聚类性能。两点间的加权欧氏距离在 $d$ 维欧氏空间可定义为

$$D(x_i,x_j)=\left\|x_i-x_j\right\|=\sqrt{\sum_{k=1}^{d}w_k\left|x_{ik}-x_{jk}\right|^2} \tag{5}$$

其中，$w_k(k=1,2,\dots,d)$ 为权重。

在本文中，MS 新的中心点由式(3)和(5)给出

$$y_{j+1}=\frac{\sum_{i=1}^{n}x_i g\left(\sum_{k=1}^{d}w_k\left|\frac{x_{ik}-x_k}{h}\right|^2\right)}{\sum_{i=1}^{n}g\left(\sum_{k=1}^{d}w_k\left|\frac{x_{ik}-x_k}{h}\right|^2\right)} \tag{6}$$

其中，在式(6)中，$h$ 为核带宽，$g(t)$ 为核函数，$w_k$ 为第 $k$ 个属性的权重系数。

由式(5)可知，权重系数对于下一个中心点的计算至关重要，对聚类性能有较大影响。由于 LFA 的特殊性，传统的 MS 算法体现出了明显的不足。鉴于此，本文对检测 LFA 的 MS 算法进行了优化，即使用加权欧氏距离替代传统的欧氏距离，采用延时率作为加权欧氏距离的权重系数。

当网络中 LFA 攻击发生后，被攻击链路负载会明显增加，所以正常数据包和 LFA 的攻击数据包时间变化量会有显著不同。其中，正常数据包时延低且分布离散，而 LFA 的数据包时延高且分布连续。

在给定的时间周期 $T$ 内，主链路中获取的数据包时延率 $T_D$ 定义为式(7)：

$$T_D=\frac{\sum_{i=1}^{n}t_{xi}-t_{xi-1}}{T} \tag{7}$$

其中，$n$ 为数据包的个数。

因此在一个时间周期 $T$ 内，网络中数据包的延时率越大，发生 LFA 的可能性越大。

### 1.2　KNN 算法原理

KNN 算法[21]基本步骤如下：

a)计算样本之间的距离；

b)将得到的未知样本和训练样本之间的距离的递增关系进行排序；

c)选取距离最小的 $k$ 个点；

d)确定前 $k$ 个点所在类别的出现频率；

e)选择出现频率最多的类别作为未知样本的类别。

KNN 算法的实现取决于未知样本和训练样本的“距离”。本文中使用的“距离”是欧氏距离，由式(5)给出

### 1.3　基于 MS-KNN 算法的 LFA 检测方法

鉴于 MS 算法以及 KNN 算法的各种优良性能，将 MS 算法与 KNN 算法相结合，得到可用于 LFA 攻击有效检测的 MS-KNN 方法。其主要步骤如下，见图 1。

a)对数据集中进行预处理，包括数据清理和标准化；

b)初始化参数；

c)将数据粗粒化，避免非常近的样本点都作为起始质心，获取可以作为起始质心的点；

d)计算均值点到每个样本点的欧氏距离与高斯核；

e)计算权重；

f)进行一次独立的均值漂移，计算下一个漂移点的坐标；

g)根据最近邻将数据分类到最近的簇中，得到 $n$ 个簇；

h)将得到的 $n$ 个簇作为输入，利用网格搜索[25]与交叉验证[26]得到每个簇使用KNN算法的最优 $k$ 值、最优权重和最优实现方法；

i)利用得到的最优KNN算法，分析每个簇，得到分析结果。

MS-KNN 的时间复杂度与空间复杂度，及其与 MS、KNN 的比较如表 1 所示。其中，$n$ 为样本数量，$T$ 为迭代次数，$k$ 为单个样本特征维度。

## 2　实验与分析

### 2.1　实验环境的构建

首先本文根据文献[2]设计了一个小型网络，如图 2 所示。该网络有 2 台配备 Intel Core i7-10700 2.90GHz 8 核处理器和 16GB 内存的计算机作为傀儡机；同时选择了 5 台配备 Intel Core i5-8400 2.80GHz 6 核处理器和 16GB 内存的计算机作为正常用户。为了保证它们之间的带宽足够高，本文选择了三台型号为 EdgeCore AS4610-54P 的千兆以太网交换机作为转发设备，这些交换机为支持 OpenFlow 协议的 SDN 交换机。控制器方面，本文选择了 RYU 控制器。本文将 RYU 控制器部署在配备 2 个 Intel(R) Xeon(R) Gold 6248R 3.00GHz 48 核处理器和 128G 内存的服务器上。该服务器被用作整个网络的控制平面，控制机器人产生攻击流量，并实现整个网络的数据收集。完成数据收集后，控制平面在数据中混合来自合法终端主机的流量特征，以生成最终数据集并进行实验分析。本文所有实验通过控制器统一分配 40 核处理器并行运算得到分析结果，其中包括数据集的聚类、网格搜索、交叉验证以及最终的数据分析。

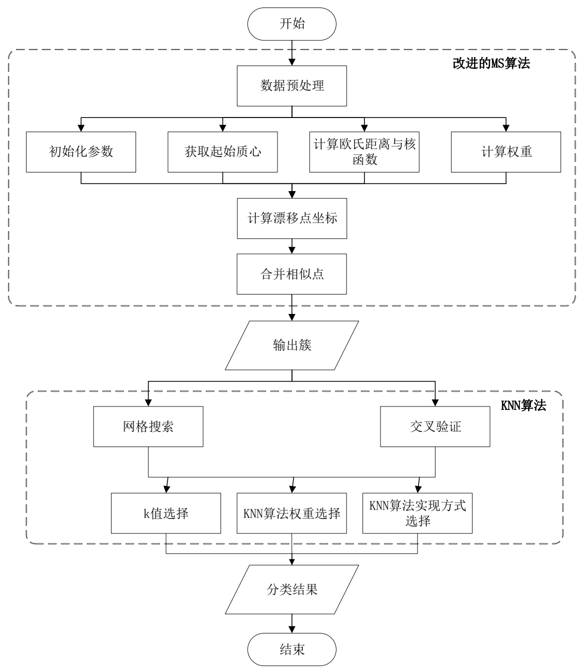



图 1　基于 MS-KNN 方法的 LFA 攻击检测流程图

Fig. 1　Flowchart of LFA attack detection based on MS-KNN method

表 1　MS、KNN 和 MS-KNN 的时间复杂度与空间复杂度

Tab. 1　Time complexity and space complexity of MS, KNN and MS-KNN

| 算法 | 时间复杂度 | 空间复杂度 |
|---|---|---|
| MS | $O(Tn^2)$ | $O(Tn^2)$ |
| KNN | $O(n*k)$ | $O(n*k)$ |
| MS-KNN | $O(Tn^2+n*k)$ | $O(Tn^2+n*k)$ |

戴尔塔式服务器（SDN 控制平面）

SDN交换机

数据收集　下发规则　关键区域　诱饵机

SDN交换机　Link Flooding Attack(LFA)　SDN交换机

傀儡机

图 2　构建的局域网拓扑图

Fig. 2　Topology diagram of the constructed LAN

### 2.2　数据集

来自合法终端主机的流量:CIC-IDS2017[27]是加拿大网络安全研究所构建的能够可靠测试和验证的数据集，如表 2 所示。

表 2　CIC-IDS2017 信息

Tab. 2　Information for CIC-IDS2017

| 日期 | 活动描述 | 攻击类型 |
|---|---|---|
| 星期一 | 正常活动 | 无 |
| 星期二 | 攻击、正常活动 | 蛮力攻击 |
| 星期三 | 攻击、正常活动 | DDoS 攻击 |
| 星期四 | 攻击、正常活动 | Web 攻击 |
| 星期五 | 攻击、正常活动 | DDoS 攻击 |

该数据集包含良性和最新的常见攻击，类似于真实的真实世界数据 (PCAP)。它还包括使用 CICFlowMeter 和基于时间戳、源和目标 IP、源和目标端口、协议和攻击(CSV 文件)标记流的网络流量分析结果。考虑到它不包含 LFA 流，本文只使用它的合法流，提取星期一合法终端主机的特征集，并将其标记为正。

傀儡机启动 LFA 的流量:由于 LFA 到目前为止没有公共数据集，所以本文基于论文[2-3]模拟 LFA 来构建该数据集。200s 内具有 LFA 流量的网络状态如图 3 所示。在 20s-80s 时，网络遭受 LFA，此时的网络吞吐量增加，波动明显。80s-135s，LFA 攻击流量减少，网络处于较安全状态，此时网络吞吐量减少且波动不大。135s-185s，LFA 攻击流量增加，网络吞吐量再次增加且波动较大。185s 之后出于安全状态，网络缓慢恢复。

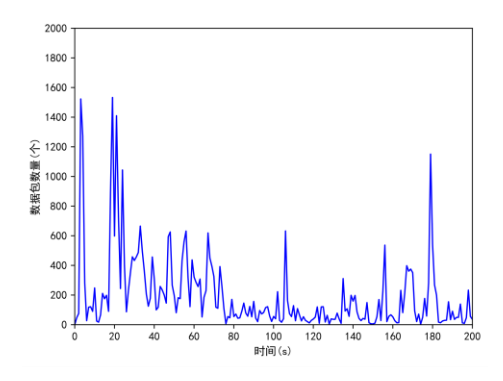



图 3　数据集中 200s 内包含 LFA 流量的网络吞吐量状态

Fig. 3　Network throughput status containing LFA traffic within 200s of the dataset

## 2.3　实验结果与分析

为了使用不同的数据结构和不同的权重实现 KNN 算法对该数据集进行初步评估，本文将数据集按照 70%/30%的比例划分为训练集和测试集，通过 TPR、FPR、PPV 和 ACC 4 个指标进行对比，结果如图 4 所示。其中，图 4(a-c)的数据显示了在统一权重下分别用三种方法实现 KNN 算法对数据的评估。实验结果表明，图 4(b)和图 4(c)的效果优于图 4(a)。图 4(b)的数据显示了当在统一权重下使用球树方法[28]，当 k 值为 10 时，此时的 TPR、PPV、ACC 分别达到了 98.92%、96.98%、96.75%，FPR 降低到了 10.99%；图 4(c) 的数据显示了当在统一权重下蛮力实现，当 k 值为 3 时，TPR、PPV、ACC 分别达到了 79.66%、96.89%、94.98%，FPR 降低到了 0.72%。相较于图 4(b)，图 4(c)虽然在 TPR、PPV、ACC 这三个指标上略有不足，但具有更低的 FPR。图 4(d-f)的数据显示了在距离倒数的权重下分别用三种方法实现 KNN 算法对数据的评估。实验结果表明，图 4(d-f)都具有较高的 TPR、PPV、ACC。图 4(d)的数据显示了当在距离倒数权重下使用 kd 树方法[29]；当 k 值为 19 时，TPR、PPV、ACC 分别达到了 98.57%、97.23%、96.69%，FPR 降低到了 10.01%；图 4(e)的数据显示了当在距离倒数权重下使用球树方法，k 值为 15 时，此时的 TPR、PPV、ACC 分别达到了 89.92%、89.79%、95.55%，FPR 降低到了 2.87%；图 4(f)的数据显示了当在距离倒数权重下使用球树方法，当 k 值为 49 时，TPR、PPV、ACC 分别达到了 97.06%、97.19%、95.51%，FPR 降低到了 10.01%。相较于图 4(d)和图 4(f)，图 4(e)虽然在 TPR、PPV、ACC 这三个指标上略有不足，但具有更低的 FPR。

$$TPR = \frac{\text{正类被分类为正类样本数}}{\text{正类样本总数}} \tag{8}$$

$$FPR = \frac{\text{负类被分类为正类样本数}}{\text{负类样本总数}} \tag{9}$$

$$PPV = \frac{\text{正类被分类为正类样本数}}{\text{被分类为正类样本数}} \tag{10}$$

$$ACC = \frac{\text{分类正确的样本数}}{\text{样本总数}} \tag{11}$$

表 3 给出了使用不同的数据结构和不同的权重实现 KNN 算法对该数据集进行初步评估的时间消耗，本文发现使用 kd 树实现所需的时间最少，蛮力实现所需的时间最多，球树介于两者之间。

数据显示，仅仅使用 KNN 算法分析数据集，虽然对于检测 LFA 具有较好的效果，然而会带来较高的 FPR，而且检测需要很长的时间。为了降低 FPR 以及检测时间，本文利用 MS-KNN 方法首先对数据集进行聚类处理，将整个数据集划分为多个簇，减少数据量从而减少检测时间；然后对每个簇使用网格搜索和交叉验证选取最优参数；最后利用最优参数分析每个簇，得到最优结果，降低 FPR。同时为了减少数据集对实验结果的影响，本文去掉了源地址、目的地址等相关特征，用协议、包长度等特征进行实验。

通过网格搜索和不同交叉验证法获得每个簇的最优参数。结果表明无论是二折交叉验证法，五折交叉验证法，还是十折交叉验证法，网格搜索的最优 KNN 算法的实现方法都是球树方法，权重基本都是统一权重，仅在十折交叉验证法下的第一个簇的权重为距离的倒数。对应的在不同交叉验证法下，每个簇消耗的时间如图 5 所示。因此，本文使用的最优参数，权重选择为统一权重，实现方法选择球树方法。

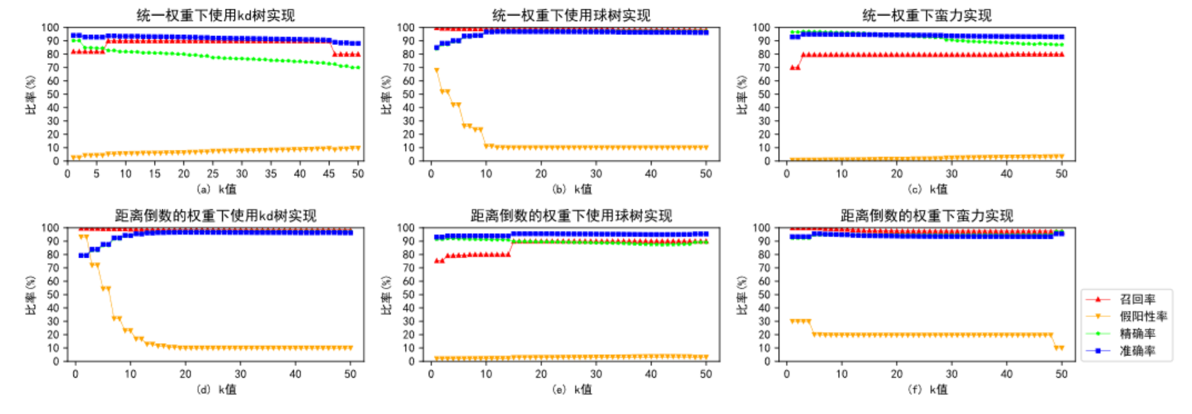



图 4　分别在统一权重和距离倒数权重下使用 kd 树、球树和蛮力方法实现 KNN 算法对数据的初步评估

Fig. 4　Initial evaluation of data by KNN algorithm using kd tree, ball tree and brute force methods under uniform weight and distance inverse weight, respectively

表 3　KNN 算法不同实现方式处理数据的时间比较

Tab. 3　Comparison of the time to process data for different implementations of the KNN algorithm

| 权重 | 方法 | 时间/s |
|---|---|---|
| | kd 树 | 59.84 |
| 统一权重 | 球树 | 117.99 |
| | 蛮力实现 | 722.22 |
| | kd 树 | 49.58 |
| 距离的倒数 | 球树 | 104.48 |
| | 蛮力实现 | 715.32 |

图 6 和 7 显示了 MS-KNN 方法对数据集的最终检测效果。图 6 给出了各个簇的 4 个评价指标，TPR、PPV 和 ACC 均达到了 99%以上，FPR 降低到了 1%一下，其中 TPR、PPV 和 ACC 最高分别达到了 99.99%、99.95%、99.98%，FPR 最低达到了 0.05%。图 7 给出了各个簇使用 MS-KNN 的检测时间，相较于表 1 中 KNN 算法对数据集的处理时间，MS-KNN 大大减少了检测所需要的时间。综合数据表明，相较于传统的 KNN 算法，MS-KNN 方法在用于 LFA 检测方面不仅取得了更高的 TPR、PPV 和 ACC，以及更低的 FPR，而且大大的减少了时间开销。

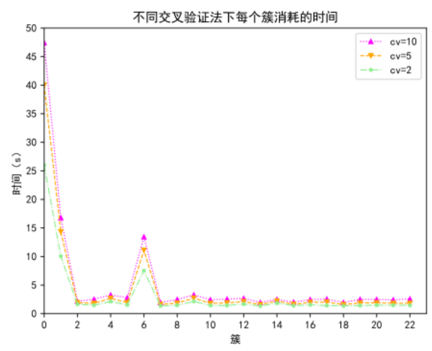



图 5　二折交叉验证法，五折交叉验证法和十折交叉验证法下每个簇消耗的时间

Fig. 5　Time consumed per cluster under two-fold cross-validation, five-fold cross-validation and ten-fold cross-validation methods

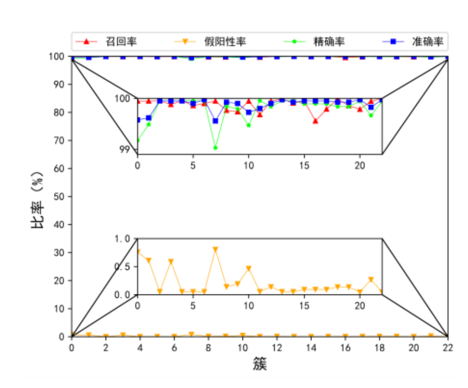



图 6　每个簇在最优参数下使用 MS-KNN 的最优评估

Fig. 6　Optimal evaluation of each cluster under optimal parameters using MS-KNN

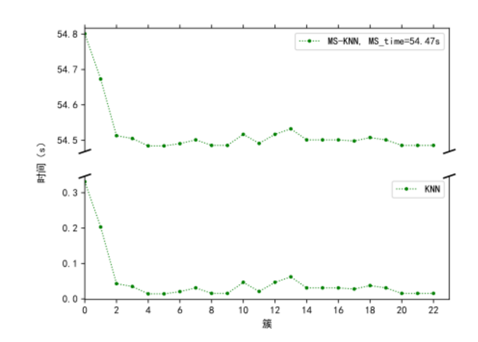



图 7　各个簇使用 MS-KNN 的检测时间

Fig. 7　Detection time of each cluster using MS-KNN

## 3　结束语

针对关键链路的链路泛洪攻击会造成链路拥塞和目标网络区域断开等严重危害。本文提出了 MS-KNN 方法，该方法将传统的 MS 算法的欧氏距离变换为加权欧氏距离，利用数据包延时率作为加权欧氏距离的加权系数来优化聚类性能，提高了聚类效果。此外，通过网格搜索和交叉验证法选取 KNN 算法的最优参数来分析 LFA 数据集。实验结果表明，MS-KNN 方法对于检测 LFA 数据集不仅有较高的 TPR、PPV 和 ACC，而且还有较低的 FPR 和检测时间。

本文提出的方法在针对 LFA 检测中取得了良好的效果，但是还存在一些不足：在未来的工作中，将构建更加复杂的网络环境，在更加真实的物理环境中研究全面的网络流量信息特征。同时，还将致力于研究如何实时动态的检测网络流量，进一步验证 LFA 检测方法的实用性。

## 参考文献：

[1] 绿盟科技. 2021DDoS 攻击态势报告 [R/OL]. (2022-02-09) . https://www. nsfocus. com. cn/html/2022/92_0209/173. html.

[2] Studer A, Perrig A. The coremelt attack [C]// European Symposium on Research in Computer Security. Berlin, Heidelberg: Springer, 2009: 37-52.

[3] Kang M S, Lee S B, Gligor V D. The crossfire attack [C]// IEEE symposium on security and privacy. [S. I. ] : IEEE, 2013: 127-141.

[4] Kang M S, Gligor V D, Sekar V. SPIFFY: Inducing Cost-Detectability Tradeoffs for Persistent Link-Flooding Attacks [C]// [S. I. ] : NDSS. 2016, 1: 53-55.

[5] Csikor L, Szalay M, Rétvári G, *et al.* Transition to SDN is HARMLESS: Hybrid architecture for migrating legacy ethernet switches to SDN [J]. IEEE/ACM Transactions On Networking, 2020, 28 (1): 275-288.

[6] Chica J C C, Imbachi J C, Vega J F B. Security in SDN: A comprehensive survey [J]. Journal of Network and Computer Applications, 2020, 159: 102595.

[7] Isyaku B, Mohd Zahid M S, Bte Kamat M, *et al.* Software defined networking flow table management of openflow switches performance and security challenges: A survey [J]. Future Internet, 2020, 12 (9): 147.

[8] 夏计强, 崔鹏帅, 李子勇, 等. 基于 P4 的 SDN 控制-数据平面流规则一致性校验 [J/OL]. 计算机应用研究: 1-6 [2022-03-30]. DOI: 10. 19734/j. issn. 1001-3695. 2021. 12. 0694. (Xia Jiqiang, Cui Pengshuai, Li Ziyong, *et al.* P4-based rules consistency verification for SDN control-data plane. [J/OL]. Application Research of Computers: 1-6 [2022-03-30]. DOI: 10. 19734/j. issn. 1001-3695. 2021. 12. 0694.)

[9] Singh J, Behal S. Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions [J]. Computer Science Review, 2020, 37: 100279.

[10] 贾锟, 王君楠, 刘峰. SDN 环境下的 DDoS 检测与缓解机制 [J]. 信息安全学报, 2021, 6 (01): 17-31. DOI: 10. 19363/J. cnki. cn10-1380/tn. 2021. 01. 02. (Jia Kun, Wang Junnan, Liu Feng. DDoS detection and mitigation Framework in SDN. [J]. Journal of Cyber Security, 2021, 6 (01): 17-31. DOI: 10. 19363/J. cnki. cn1 0-1380/tn. 2021. 01. 02.)

[11] Rezapour A, Tzeng W G. Rl-shield: mitigating target link-flooding attacks using sdn and deep reinforcement learning routing algorithm [J]. IEEE Transactions on Dependable and Secure Computing, 2021.

[12] Ravi N, Shalinie S M, Theres D D J. BALANCE: link flooding attack detection and mitigation via hybrid-SDN [J]. IEEE Transactions on Network and Service Management, 2020, 17 (3): 1715-1729.

[13] Wang J, Wen R, Li J, *et al.* Detecting and mitigating target link-flooding attacks using SDN [J]. IEEE Transactions on Dependable and Secure

Computing, 2018, 16 (6): 944-956.

[14] Liaskos C, Ioannidis S. Network topology effects on the detectability of crossfire attacks [J]. IEEE Transactions on Information Forensics and Security, 2018, 13 (7): 1682-1695.

[15] Xue L, Ma X, Luo X, *et al.* Linkscope: Toward detecting target link flooding attacks [J]. IEEE Transactions on Information Forensics and Security, 2018, 13 (10): 2423-2438.

[16] Wang L, Li Q, Jiang Y, *et al.* Woodpecker: Detecting and mitigating link-flooding attacks via SDN [J]. Computer Networks, 2018, 147: 1-13.

[17] Kim J, Shin S. Software-defined HoneyNet: Towards mitigating link flooding attacks [C]// The 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W) . [S. I. ] : IEEE, 2017: 99-100.

[18] Deb R, Roy S. A comprehensive survey of vulnerability and information security in SDN [J]. Computer Networks, 2022: 108802.

[19] Hande Y, Muddana A. A survey on intrusion detection system for software defined networks (SDN) [M]// Research Anthology on Artificial Intelligence Applications in Security. IGI Global, 2021: 467-489.

[20] Li W, Meng W, Liu Z, *et al.* Towards blockchain-based software-defined networking: security challenges and solutions [J]. IEICE Transactions on Information and Systems, 2020, 103 (2): 196-203.

[21] Zhang S. Cost-sensitive KNN classification [J]. Neurocomputing, 2020, 391: 234-242.

[22] Zhang Y, Chen Y C. Kernel smoothing, mean shift, and their learning theory with directional data [J]. Journal of Machine Learning Research, 2021, 22 (154): 1-92.

[23] Huang K, Fu X, Sidiropoulos N. On convergence of epanechnikov mean shift [C]// Proceedings of the AAAI Conference on Artificial Intelligence. 2018, 32 (1) .

[24] Silverman B W. Density estimation for statistics and data analysis [M]. Routledge, 2018.

[25] LeCun Y, Bengio Y, Hinton G. Deep learning [J]. nature, 2015, 521 (7553): 436-444.

[26] Marcot B G, Hanea A M. What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis? [J]. Computational Statistics, 2021, 36 (3): 2009-2031.

[27] Sharafaldin I, Lashkari A H, Ghorbani A A. Toward generating a new intrusion detection dataset and intrusion traffic characterization [J]. ICISSp, 2018, 1: 108-116.

[28] Cui L, Zhang Y, Zhang R, *et al.* A modified efficient KNN method for antenna optimization and design [J]. IEEE Transactions on Antennas and Propagation, 2020, 68 (10): 6858-6866.

[29] Xu Y, Yu Y, Hong H, *et al.* DDoS detection using a cloud-edge collaboration method based on entropy-measuring SOM and KD-tree in SDN [J]. Security and Communication Networks, 2021, 2021.